Desarrollo de cámara multiespectral: captura y análisis de sus imágenes

Luis Lebron Casas

Resumen– Para realizar reconocimiento y clasificación de imágenes tradicionalmente se han usado imágenes en escala de grises o imágenes en color iluminadas con luz dentro del espectro visible, de los 400 nm a los 700 nm, limitando la información que puede captar una máquina fuera de este rango de frecuencias. El propósito de este proyecto es estudiar el efecto que provoca las utilización de imágenes multiespectarles en la clasificación de objetos.El primer objetivo del proyecto es acondicionar una cámara adaptando una iluminación multiespectral más barata que las que se comercializan actualmente. Esta cámara se compone de un arduino, una cámara en blanco y negro y diversos LED de diversas longitudes de onda. La segunda parte del proyecto consiste en el estudio y comparativa de diversos algoritmos de clasificación en diversos espacios de color y con distintas características, basándose en la salida que nos entrega esta cámara multiespectral.

Palabras clave– Imágenes, Multiepectral, Support Vector Machine, Random Forest, Convolutional Neuroal Network, Local Binary Pattern, Scale-Invariant Feature Transform, DAISY, RGB, LAB, Arduino

Abstract– Traditionally the image classification and recognition have been done using grey or color images in the boundaries of the human eye, which is between the 400 mm and the 700nm, these limits produce a reduction of the information which a machine can perceive. The objective of this project is to study the effect of these images on the object's clasificación problem. The first goal of the project is to adapt a multiespectral camera cheaper than the ones that are being sold nowadays. This camera will be made using one Arduino, a black and white camera and some LED of different spectrums. The second part will consist in the study and comparison of some clasificación's algorithms in different color spaces and feature descriptors, giving the output of the multiespectral camera.

Keywords– Imagen, Multiepectral, Hyperespectral, Support Vector Machine, Random Forest, Convolutional Neuroal Network, Local Binary Pattern, Scale-Invariant Feature Transform, DAISY, RGB, LAB, Arduino

1 Introduction

PARTIENDO de un problema de clasificación muy complejo, clasificar piedras de riñón, y buscando nuevas fuentes de información que mejorasen esta clasificación se ha optado por explorar el uso de iluminación con múltiples longitudes de onda. Extendiendo la típica información RGB más allá del rango visible.

La forma tradicional de clasificar las piedras de riñón es encontrar de qué materiales están hechas, para realizar esta tarea se pueden usar métodos químicos o métodos más comunes como observar el color o la textura. Por tanto la forma de resolver este problema de forma automática sería analizando las imágenes e intentar diferenciarlas por el color principalmente y la textura secundariamente.

En una primera toma de contacto con el problema de clasificar piedras de riñón, se observó que con imágenes iluminadas con una luz blanca normal no se conseguía el resultado esperado dado que clases muy diferentes tienen un aspecto tan similar que no las podía distinguir.

En la actualidad las cámaras multiespectrales están tomando un papel muy decisivo en muchos problemas: análisis de cultivos, clasificación de materiales o investigación forense. Queremos acercarnos a este problema

• E-mail de contacte: luis.lebron@e-campus.uab.cat.
• Menció realitzada:Computació
• Treball tutoritzat per: Felipe Lumbreras (CVC)
• Curs 2015/16

por un lado para ver hasta dónde podemos llegar con un diseño simple y por otro explorando las posibilidades de construcción de un dispositivo de bajo coste dentro de este ámbito.

1.1 Motivación

La motivación de este proyecto es desarrollar una cámara multiespectral y haciendo uso de ésta estudiar formas de clasificación de imágenes. La principal motivación para elegir este proyecto es el poder aprender a trabajar con otro tipo de imágenes a la vez que con una tecnología nueva y poder aprender sobre herramientas para clasificarlas.

1.2 Objetivos

El objetivo principal del proyecto es desarrollar una cámara multiespectral, o un método de captación de imágenes multiespectral para posteriormente probar su funcionamiento con tareas de clasificación.

Estas tareas de clasificación empezarán con problemas simples y acotados. No llegaremos a la clasificación de muestras de riñón debido a su complejidad pero si que daremos los primeros pasos.

1.3 Metodología de trabajo

Dado que el equipo de este proyecto solo consta de una persona y se desea tener una metodología en que se acepten cambios y revisiones, el tipo de metodología que se ha elegido ha sido un tipo AGIL. El método de trabajo que he escogido ha sido SCRUM [1] puesto que es muy conocido y fácil de aplicar. Dado que no es un equipo de varias personas no se podrán aplicar todas sus características pero siguiendo las recomendaciones se puede conseguir los resultados deseados al tener un control por sprints, y también sobre los resultados de cada fase. La duración de los sprints ha sido semanal con revisión del estado en ese momento y poder adaptar el siguiente sprint en base al desarrollo conseguido.

2 REQUISITOS

2.1 Requisitos funcionales

Los requisitos funcionales de este proyecto son:

- Observar si se puede mejorar una clasificación al usar imágenes multiespectral.
- Usar código libre para mantener costes razonables, y poder configurarlo siempre que se pueda según nuestras necesidades.
- La correcta captación de las imágenes por la cámara.
- Tiene que ser fácil de configurar y mostrar el máximo de información posible.

2.2 Requisitos hardware

En el apartado hardware se ha elegido usar una cámara Basler Dart daA2500-14um [2] y 16 LED diferentes para la iluminación.

Los requisitos hardware mínimos vienen dados por el software utilizado. El objetivo es que los requisitos sean los mínimos para facilitar su uso en cualquier ordenador convencional.

2.3 Requisitos software

El sistema operativo elegido es Windows por el hecho de que tiene un mayor número de usuarios y el segmento que se enfoca el sistema usa este tipo de plataforma. Más concretamente se usará Windows 10. El software utilizado en este proyecto se puede dividir en dos partes: una dedicada a la captación de las imágenes con la cámara y otra a la clasificación de estas imágenes.

La parte de captación de imágenes se ha hecho con un programa desarrollado en C++ compilado sobre Visual Studio 2010 con las librerías propias de la cámara y su propia API, además de una librería para poder guardar las imágenes juntas en un formato adecuado. El formato elegido es TIFF dada su flexibilidad y las diversas configuraciones optativas que dispone [3].

La segunda parte del proyecto, la clasificación de imágenes será hecha en Matlab porque acepta las imágenes multiespectarles a la vez que cuenta con una amplia disposición de librerías para el tratamiento y clasificación de imágenes.

3 PLANIFICACIÓN

El proyecto está dividido en dos partes centrales referentes al tema tratado, y otras dos al principio y final con el objetivo de preparar la documentación. En general el proyecto se realizará en 300 horas. Para la parte dedicada a la captura de las imágenes se han dedicado alrededor de 120 horas y 150 horas se han destinado a las pruebas de las diferentes tareas de clasificación. El resto de horas han servido para hacer la documentación requerida. Los detalles de la planificación están reflejados en el diagrama de gantt que se encuentra en el apéndice A.2.

4 RIESGOS DEL PROYECTO

En esta sección se detallarán los diferentes riesgos que pueden surgir durante la realización del proyecto y cómo se ha planteado solucionarlos o qué medidas tomar en el caso de darse.

Uno de los riesgos mas importante de este proyecto es que la cámara no funcione como esperamos, esto provocaría que se tuviera que modificar partes o rediseñar completamente la cámara. Otro riesgo destacable sería que las muestras no fueran suficientes para llevar acabo los diferentes experimentos, en este caso la solución sería cambiar los materiales a clasificar por unos de los que sí tengamos muestras suficientes. En la tabla en el apéndice A.1 está detallado todos estos casos.

5 ESTADO DEL ARTE

Aunque el tema principal de este trabajo son las imágenes multiespectral también se tratan otros temas como son los

algoritmos de clasificación. En esta sección se hablará de cuál es el estado actual de todas estas áreas.

5.1 Imágenes multiespectral

Hasta ahora este campo de estudio sólo se había usado para el análisis de cuerpos celestes o geología, dado que las cámaras multi/hiperespectrales eran muy caras. El principal objetivo de este tipo de imágenes es poder detectar ciertos componentes que sólo se ven en ciertas bandas del espectro de luz o que tienen un mayor valor numérico en bandas concretas.

Actualmente gracias al abaratamiento de esta tecnología se ha ampliado su campo de uso para hacer análisis en varias áreas como son el control de los campos de cultivos, o el análisis cualitativo de comida o fármacos. [4]

5.1.1 Introducción a las imágenes multiespectarles

Aunque el ojo humano está limitado a percibir una serie única de colores, existen muchas más bandas que una cámara puede percibir. Las imágenes multiespectral contienen más bandas que las que se perciben en el espectro visible.

La principal diferenciación entre las imágenes multiespectarles y las hiperescpectrales es el número de bandas que utilizan. En el caso de las multiespectrales contienen pocas bandas y en cambio en las imágenes hiperespectrales agrupan cientos de bandas permitiendo observar mejor los detalles de cada banda, pero tiene la desventaja tanto tener un coste elevado como la dificultad de su obtención y análisis debido también a su tamaño.

5.1.2 Diferentes tecnologías para la obtener imágenes multiespectral

Hay cuatro tipos principales de formas de adquirir una imagen multiespectral: “Spatial scanning”, “Spectral scanning”, “Non-scanning” y “Spatioepectral scanning”. [5]

El primero consiste en proyectar una escena de forma que quede dividida y dispersada según las bandas que se pueden captar en la imagen. Después se captura esta imagen con una cámara que se desplaza linealmente, llamada push broom scanner, y se analiza línea a línea. La mayor desventaja de este sistema es que tiene una cierta complejidad en la cámara debido a que la toma se hace en movimiento y solo analiza las líneas.

El siguiente tipo es el que utiliza la cámara resultante de este proyecto y consiste en generar varias imágenes 2D monocromática, cada una de estas representa la escena en una banda diferente. El problema de estas cámaras es el tiempo requerido para conseguir la imagen que conlleva la necesidad de mantener la escena quieta durante la captura.

El tipo “Non-scanning” es similar al anterior en el hecho de que montan un cubo con todos los espectros representados por imágenes monocromáticas 2D. Pero a diferencia del anterior este captura la escena en una sola toma. La desventaja de este caso es el coste y la complejidad computacional que requiere.

En el último tipo de cámara multiespectral el objetivo es solucionar las desventajas de los tipos “Spatial scanning” y “Spectral scanning” al unirlos. En este método la salida del sensor es una representación 2D de la escena con la longitud de onda mapeada en la imagen. Por lo que cada posición x,y representa la descomposición de colores a la vez que describe el objeto.

5.2 Clasificadores

Aunque hay una gran variedad de algoritmos para realizar este tipo de clasificaciones, en este proyecto solo se usarán los algoritmos de aprendizaje inductivo supervisado porque son los que mejor se adaptan a la definición del problema que deseamos resolver. Estos algoritmos definen dos fases claramente, una de entrenamiento y otra de pruebas. En la fase de entrenamiento se necesita una información externa la cual indique la clase o etiqueta de la muestra que se está aprendiendo, por lo que se necesita tener catalogadas todas las muestras de esta fase. En la siguiente fase, la de pruebas, se evalúa su comportamiento ante muestras desconocidas para el algoritmo pero que los resultados sean conocidos para el método de evaluación. También permiten añadir unas últimas fases para validar el resultado final.

5.2.1 Redes Neuronales

Uno de los tipos de clasificadores que dan mejor resultado son las redes neuronales convolutivas [6]. Este tipo de algoritmo de aprendizaje tiene la facilidad de que no depende tanto del conocimiento previo que se tiene de la materia sino que solo requieren saber para cada muestra cual es el resultado esperado, y es el propio algoritmo el que extrae las características que usará después para clasificar.Esto facilita la tarea de programación de la red neuronal pero también tienen la desventaja que necesitan muchas muestras y mucho tiempo de aprendizaje.

Este algoritmo consiste en simular el comportamiento de las neuronas a través de un proceso informático representado por un grafo, así cada nodo de información se activa con un valor según el estímulo de entrada. Para conseguir los factores que deciden el valor de activación de cada nodo se necesita una fase de entrenamiento en la cual consiste en entrenar la red con parejas de muestra-resultado y la red va adaptando sus valores internos.

5.2.2 Otros algoritmos

Por tal de comparar como interactuan las muestras con otros métodos de clasificación se usarán los algoritmos Random Forest, RF, y Support Vector Machine, SVM. Estas dos formas de clasificar pueden necesitar que los datos estén pretratados por tal de mejorar su resultado por ello también se añadirán diversos espacios de color y diversas características.

El Support Vector Machine es un algoritmo que intenta separar las muestras definiendo vectores que separan las clases. Ante una nueva muestra calcula su posición respecto a estos vectores para determinar a qué clase pertenecen. [8]

El algoritmo de Random Forest consiste en usar varios decision trees [9]. Para una nueva muestra, cada uno de los árboles hacen una clasificación y se elige la que más haya aparecido. Los árboles son entrenados de forma que crezcan al máximo posible sin hacer ninguna poda. [10]

5.3 Caracteristicas

5.3.1 Espacios de color

En referente a las comparaciones con otros espacios de color se han usado varios de diferentes, estos serían: RGB, PCA para reducir el número de bandas, Color Naming y LBA.

El primer espacio de color aparte de las 15 bandas de la captura de imagen es el Principal Component Analysis o PCA [7]. En este caso se intenta reducir la complejidad del problema reduciendo el número de bandas seleccionando las que aportan más información por esto se toma como referencia una matriz de coeficientes que representan la varianza de cada vector respecto al espacio. La matriz resultante contiene como primer vector el que representa la mayor varianza en el espacio original y así sucesivamente. Con esta matriz se eligen las 3 bandas mas significativas y se crea una imagen de 3 bandas con ella.

En el siguiente espacio, el RGB, se ha usado las bandas que representan sus colores creando un RGB artificial. Los colores seleccionados como rojo, el LED de 700nm, como verde el LED 525nm, y como azul el LED de 470. Es comprensible que en este modo haya una perdida de calidad respecto al espacio de 15v bandas o a la PCA ya sea por falta de información o porque no son los canales que aportan más información.

El RGB se convierte en otros dos espacios de colores: LAB y Color Naming. Como LAB se utiliza CIE 1976 L*a*b*, LAB es un espacio de color que tiene el propósito de conseguir una percepción mas líneal, más similar al del ojo humano. En el caso del Color Naming se busca reducir la complejidad de la imagen usando el nombre del color para describir a la zona de referencia.

5.3.2 Detectores de características

Como descriptores de características se utilizan tres: LBP, SIFT y DAISY. Con estos descriptores se intenta reducir el volumen de información dada por cada canal o detectar características que solo se hallan en algunos canales.

El descriptor LBP (Local binary patterns) [11] consiste en dividir la imagen en ventanas mas pequeñas, comparar un número limitado de pixeles vecinos en cada ventana, según si el valor del pixel es mayor a los de sus vecinos se escribe 1 o 0, de este resultado se extrae un histograma. Los histogramas de todos los pixeles de la ventana determinan el vector de características de la zona.

SIFT (Scale-invariant feature transform) [12] consiste en un histograma 3D en el espacio de puntos significativos. Cada pixel se describe con un gradiente que es representado por unos vectores tridimensionales que dan la posición del pixel y la orientación del gradiente. Dentro de una zona se normaliza estos gradientes y se genera un histograma que representa las características de la zona. Por tal de eliminar gradiente que aportan poca información se usa una gaussiana respecto al centro del punto significativo para penalizar la distancia. EL SIFT utilizado es denso por lo que se usan un mallado de puntos por toda la imagen como puntos significados.

El último descriptor es DAISY [13] que igual que el SIFT utilizado es denso. DAISY es un descriptor basado en histogramas de orientación de gradientes como SIFT pero a diferencia de este DAISY utiliza una distribución de gaussianas en círculos respecto al punto para describirlo.

6 Funcionamiento de la captura de imágenes

En este apartado se describirá la estructura de la cámara y los elementos de que esta compuesta.También se explicará como funciona la cámara, que secuencia sigue para tomar la imagen y los posibles parámetros de configuración que se pueden utilizar.

6.1 Estructura de la cámara

La estructura de la cámara consiste en 15 LED de diferentes bandas puesto en círculo para iluminar una zona central y una cámara en blanco y negro que captura la imagen. La cámara esta situada por encima de los LED y enfoca al área central que será iluminada por los LED. La posición de los componentes se puede observar en la imagen 1. Como se puede observar el objeto al que queremos tomar la imagen se debe situar en el centro del círculo de LED.

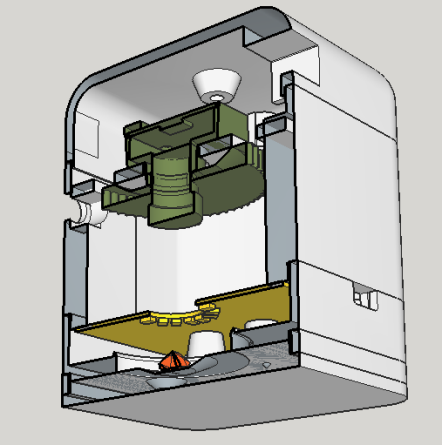

Figure 1: Diagrama de la estructura de la cámara

La cámara usada, como se ha mencionado en apartados anteriores, es una Basler Dart daA2500-14um que es una cámara en blanco y negro. Referente a los LED se utilizan 15 LED con diferentes longitudes de onda. La longitudes son: 395nm (Banda 1), 430nm (Banda 2), 470nm (Banda 3), 525nm (Banda 4), 630nm (Banda 6), 680nm (Banda 7), 700nm (Banda 8), 735nm (Banda 9), 750nm (Banda 10), 770nm (Banda 11), 810nm (Banda 12), 850nm (Banda 13), 890nm (Banda 14), 940nm (Banda 15) y 970nm (Banda 16). Un ejemplo de cada banda se puede observar en la figura 2. La cámara permite 16 LED pero en esta configuración solo se utilizan 15, faltando el que estaría en la posición 5. Como se puede observar en la figura 2 a la hora de montar una imagen RGB a partir de las capturas de la cámara surgen problemas debido a que la luz no es uniforme para toda la imagen y cada LED ilumina con más claridad el área más cercana a él. En el centro de la imagen es donde se puede apreciar mejor los colores verdaderos de la escena.

La posición de los LED está detallada en la imagen 3. Por tal de controlar que LED están encendidos en cada

momento se utilizan 15 salidas,tanto analógicas como digitales, de un Arduino NANO para controlar la secuencia de LED. La sincronización entre cada una de las partes de la

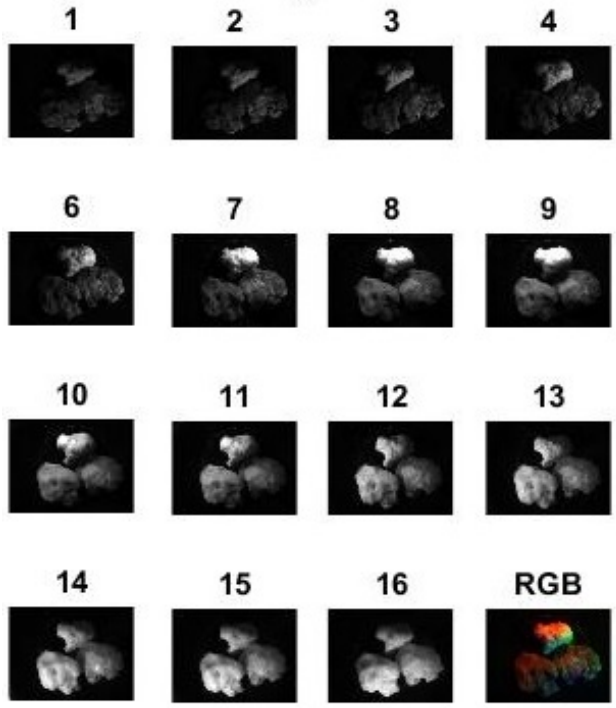

Figure 2: Ejemplo de imágenes multiespectral y RGB

cámara multiespectral está hecha con un programa en C++ que interactua con la API de la cámara y manda las instrucciones al arduino por el bus serie. Por tanto como otra parte de la estructura de la cámara se podría considerar el ordenador al que está conectada tanto la cámara Basler como el arduino. También será donde se guardaran las imágenes multiespectrales.

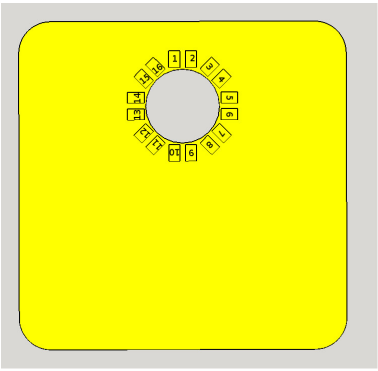

Figure 3: Diagrama de la posición de los LED

6.2 Captura de la imagen

La captura de una imagen con la cámara descrita en el apartado anterior seguirá un proceso estricto. Como se ha comentado un requisito de este tipo de cámara es que la escena tiene que mantenerse quieta durante todo el proceso. El flujo de captura sería el siguiente:

1. Se conecta el arduino y la cámara a través de los puertos serie correspondientes.
2. Se inicia el programa.
3. El programa carga la configuración en donde se le indica tanto los LED a encender como la configuración de la cámara.
4. Se enciende el LED o los LED configurados para la toma.
5. Se carga en la cámara la configuración.
6. Se realiza la captura de la imagen y se guarda la imagen por separado en un directorio.
7. Se apagan los LED encendidos.
8. Se repiten los pasos a partir del punto 4 hasta haber tomado todas las capturas definidas en el fichero de configuración.
9. Al finalizar guardamos un fichero con la configuración de salida y una imagen TIFF mulitespectral con todas las capturas.

El orden de estos pasos es estricto pero se admite flexibilidad en que LED se enciende en cada momento. También se pueden configurar varias opciones de la cámara, estas configuraciones se explicaran en la siguiente sección.

6.3 Configuraciones de la cámara

La cámara requiere dos ficheros de configuración. Uno que define parámetros sobre una captura y otro que define el wavelength de cada uno de los LED de la cámara.

El fichero que contiene los wavelengths consiste en una relación del número del LED de zero al número de LED de la cámara con el wavelength en nm de cada uno de ellos. En el otro fichero de configuración se puede definir varios parámetros. Todos los parámetros tienen que estar bien escritos ya que diferencia de minúsculas y mayúsculas pero no importa en el orden en que estén puestos. En el caso de los LED el orden entre ellos solo definen su posición en la imagen final pero la secuencia ordenada estará explicada en el fichero de configuración de salida.En este fichero también se aceptan comentarios que no serán guardados en el programa, la sintaxis del comentario es poner delante el símbolo #.

Uno de los parámetros es el nombre con el que se define la captura. Este nombre se usará para crear el directorio donde se guardarán las imágenes también se usará para identificar cada imagen siguiendo el formato ”nombre_X.png” siendo X el orden de la captura que define el LED o los LED encendidos en ese momento. Fuera del directorio se guardará un fichero llamado ”nombre”.cfg que contiene una configuración de salida construida a partir de la información dada por los ficheros de entrada pero se adaptará para que sea mas fácil de leer. También se guarda un TIFF ”nombre_multiespectral.tif” que guarda en una sola imagen tiff multisampleperpixel todas las bandas. Si no se especifica un nombre se utilizará la fecha actual como identificador y en cualquiera de los casos si ya existe el directorio se le añadirá un sufijo numérico para diferenciarlos.

En la configuración también se puede especificar si utilizar un modo automático en que la cámara va tomando todas las imágenes sin parar u otro manual en que después de

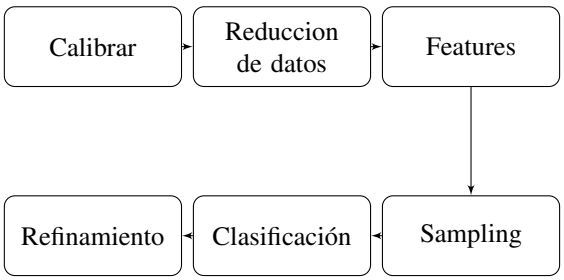

Figure 4: Pipeline de experimentos

cada captura mostrará la imagen y esperará a que el usuario pulse enter para continuar. Otro parámetro es el binning vertical y horizontal que será un número de 1 a 4 y se puede definir cada uno por separado. El binning es combinar un cluster de pixeles en uno solo así se reduce el número total de pixeles y pequeños errores. Un ultimo parámetro es el puerto de comunicación serie del arduino, si no se especifica se utilizará uno por defecto. Los últimos parámetros hacen referencia a cuales LED se enciende en cada captura y que tiempo de exposición y ganancia se utilizan. En cada captura de una toma se pueden definir de uno a más LED pero en una sola sólo se puede definir un tiempo de exposición y un ganancia.

7 EXPERIMENTOS

En esta sección se explicarán todos los experimentos realizados. Todos ellos están hechos con Matlab R2015b. El objetivo de estos experimentos es probar diferentes combinaciones de algoritmos y espacios de características para hacer una comparación entre las imágenes tomadas con la cámara multiespectral y las imágenes de uno o tres bandas.

En los siguientes apartados se tratarán las diferentes combinaciones y opciones que se aceptan en las clasificaciones en el pipeline descrito en la figura 4 y cuales han sido los resultados obtenidos.

7.1 Resultados de la captura

Después de adaptar la cámara para poder tomar imágenes multiespectrales, se probó su funcionamiento con diferentes materiales. Las primeras pruebas se hicieron con tejidos, los experimentos que se explican en las siguientes subsecciones están extraídos de estas primeras imágenes.

Pero el rango de aplicaciones de este tipo de imágenes es mayor que la de clasificación de telas o materiales por el color. Aunque este proyecto se ha enfocado mas a la clasificación de imágenes, también se ha explorado el uso de la cámara para la reconstrucción 3D usando photometric stereo. Esta idea surge al ver en las primeras imágenes unas sombras bien determinadas en las escena producida por los LED. Un ejemplo del tipo de imágenes usado para hacer esta reconstrucción serían las de la figura 5 que muestra una clara diferencia entre la sombra de cada LED.

Otro experimento que se ha intentado realizar con esta cámara ha sido el diferenciar dos tipos de arroz: uno con bichos y otro limpio. En la figura 6 se puede observan los dos tipos, siendo los granos de arriba los que contienen bichos y los de debajo los limpios. Las dos imágenes de encima son de las bandas azules y rojas, y las dos de abajo

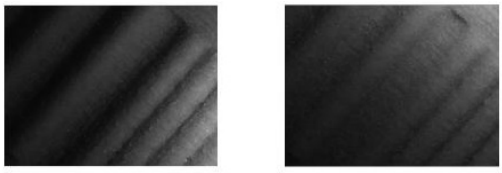

Figure 5: Dos imágenes de una figura 3D

pertenecen al espectro infrarrojo. Aparentemente no hay diferenciación en las bandas que cubre la configuración actual.

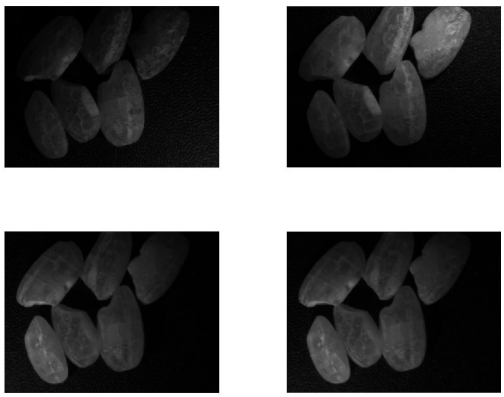

Figure 6: Tipos de arroz

Finalmente, también se han hecho tomas de imágenes de piedras pero no se ha podido hacer ningún experimento de clasificación debido al limitado número de muestras para la complejidad del experimento. Algunos de estos ejemplos se puede ver en la figura 7.

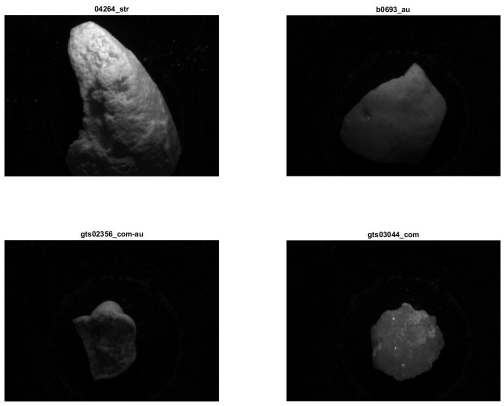

Figure 7: Cuatro tipos de piedras en la banda 630 nm

7.2 Primer Experimento

Para este primer experimento se ha usado la configuración de 10 imágenes de entrenamiento y 5 de test. Como medida de comparación se usará la precisión, total de positivos correctos entre número total de muestras, ya que lo que nos interesa es que acierte a que clase pertenece la muestra. Al ser conjuntos con múltiples clases el acertar que una muestra no pertenece a las otras clases no aporta mucha información.

El primer experimento esta realizados con 6 sets de muestras de tejidos de colores azules oscuros y grises con diferente texturas. Cada set contiene 15 muestras tomadas

con la misma configuración de binning, ganancia y tiempo de exposición. Cada imagen esta tomada en diferentes ángulos y áreas de la misma tela.

El objetivo de este experimento es probar los algoritmos de clasificación y los espacios de colores. A partir de los resultados se busca establecer unos valores de referencia para los siguientes experimentos.

7.2.1 Support Vector Machine vs Random Forest

En el caso del SVM se ha usado el modo "all pair" ya que es el que daba mejor resultado en el menor tiempo.Como se observa en la tabla 1 tanto el "one vs all" como el "all pairs" tienen los mismos resultados llegando a acertar siempre con una area de 40*40 pixeles. La diferencia entre ellos es el tiempo que tardan en aprender, en el caso de un pixel el "one vs all" tarda 0.21 segúndos en cambio el "all pair" tarda 0.14 segúndos.

	SVM one vs all	SVM all pairs	SVM gaussian
pixel	0.06	0.60	**0.30**
area=10	0.70	0.70	0.16
area=20	0.93	0.93	0.16
area=40	**1.00**	**1.00**	0.16
area=80	0.33	0.33	0.16

Table 1: COMPARATIVA DE ENTRENAMIENTOS CON SVM

En el caso del Random Forest se probó diversos números de árboles y el que daba mejor resultado era 80 árboles. En ninguno de los dos clasificadores se han variado los parámetros de número máximo de iteraciones ni el cálculo de coste. En la tabla 2 se observa los diferentes resultados variando el número de árboles. De las opciones del Random Forest solo se ha variado el número de arboles. El resto de modos o sólo eran límites al tiempo o al número de iteraciones, o eran parámetros que se deberían adaptar para cada experimento como por ejemplo la función de coste.

	RF trees=10	RF trees=20	RF trees=40	RF trees=80
pixel	0.56	0.66	0.56	0.63
area=10	0.60	0.73	0.73	0.76
area=20	**0.76**	0.80	0.90	0.86
area=40	0.73	**0.83**	0.90	**1.00**
area=80	0.70	0.73	**1.00**	1.00

Table 2: COMPARATIVA DE ENTRENAMIENTOS CON RF

7.2.2 PCA vs RGB-artificial

En este apartado se usarán los mejores resultados de los clasificadores para comparar los diferentes tipos de características. Además al reducir el número de características se pueden hacer entrenamientos con la imagen completa. En el apartado anterior no se podía usar este tipo de entrenamiento porque había demasiados datos y el clasificador tardaba demasiado.

En este experimento se utiliza una PCA para encontrar las 3 bandas mas significativas. Vemos que al principio se pierde información haciendo que los resultados sean peores y que en un cierto punto no se puede mejorar el resultado. Pero valorativamente la perdida de precisión no es tan elevada comparada a la reducción de datos.

El RGB que se ha usado para esta sección es uno construido a partir de 3 canales de la cámara multiespectral uno rojo, otro verde y un último azul de la imagen multiespectral. Como muestra los resultados de la tabla 4 se puede apreciar una mayor perdida de precisión que con la PCA dado que ahora las 3 bandas seleccionadas no son las que aportan mas información sino que su objetivo es simular una cámara RGB normal.

En la gráfica 8 se puede apreciar que la imagen multiespectral de 15 bandas aporta la suficiente información para que a partir de un punto se pueda conseguir una precisión del 100% pero en cuanto el nivel de datos es demasiado alto pierde precisión y aumenta el tiempo de aprendizaje. La perdida de precisión también se puede dar por causa de unos bordes negros en la imagen.

	SVM	RF trees=40	RF trees=80
pixel	0.50	0.56	0.56
area=10	0.63	0.76	0.70
area=20	0.93	0.8	0.86
area=40	**1.00**	0.86	0.90
area=80	0.33	**0.93**	0.93
Imagen completa	0.63	0.93	**0.96**

Table 3: COMPARATIVA DE ENTRENAMIENTOS CON PCA

	SVM	RF trees=40	RF trees=80
pixel	0.50	0.43	0.43
area=10	0.63	0.56	0.60
area=20	0.80	0.66	0.66
area=40	**0.96**	0.80	0.83
area=80	0.93	**0.90**	0.93
Imagen completa	0.63	0.90	**0.96**

Table 4: COMPARATIVA DE ENTRENAMIENTOS CON RGB-ARTIFICIAL

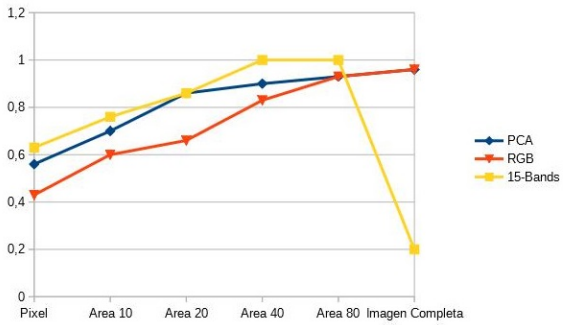

Figure 8: Comparativa RGB artificial vs PCA vs 15 Bandas

7.3 Segundo Experimento

Después de realizar el primer experimento se extrajo unas conclusións en referencia al tiempo requerido a la clasific-

ación, el número de características, y los resultados obtenidos para diseñar los siguientes experimentos.

Aunque los resultados daban una precisión alta estos requerían un gran volumen de características que llevaban a largos tiempos para procesar y clasificar las muestras. Esta primera conclusión llevó a preguntarnos si eran necesarias tantas características para hacer la clasificación. Además también se aumenta el número de descriptores a utilizar por tal de probar si es mejor una clasificación por características en los diferentes canales de color o hacer una clasificación simple por color.

En este experimento se usaron cuatro panas de diferentes color: dos marrones oscuros, una azul oscuro y una negro. De cada una ellas se tomaron 30 imágenes haciendo rotaciones y desplazamientos de la propia cámara. La configuración de la cámara se cambió eliminando la ganancia y aumentando el tiempo de exposición. Además se usaron tres tiempos de exposición diferentes por cada banda por tal de estudiar si era mejor usar el tiempo de exposición extraído de la calibración manual de la cámara, o la mitad de este para que los blancos no lleguen a saturar o el doble por tal de poder diferenciar mejor los colores oscuros. Por tanto, cada imagen cuenta con 45 bandas. Además para evitar problemas con los bordes oscuros o la saturación en donde está el LED se ha escogido trabajar sobre una área de 40x40 en el centro de la imagen por ser el sitio donde los colores están mejor calibrados.

7.3.1 RGB vs PCA vs CNaming vs 15-bandas

En la figura 9 se observa la comparativa entre los diferentes espacios de color, cada prueba se ha hecho con una área de 40 por 40 con la imagen centrada. Se puede observar que ya que el RGB tiene diferencias con una imagen RGB captada con una cámara corriente esto perjudica su resultado. Los espacios extraídos de este RGB, el CNaming y el Lab también añaden error a su clasificación. CNaming sale perjudicado debido a que depende de la calibración previa de la cámara para determinar el nombre del color. En el caso de LAB tienen el inconveniente de que codifica la luminiscencia pero con esta cámara y la calibración manual no se ha podido mantener el mismo nivel en todas las bandas. En espacios como LAB se puede ver la diferencia entre algoritmos que utilizan toda la información dada,como el SVM, y los que son mas flexibles al elegir que información usarán,como es Random Forest.

Aún dado estos problemas con la comparación de la PCA se extrae que puede hacer una clasificación con una alta precisión y que los mejores resultados provienen de las imágenes multiespectral con 15 bandas.

7.3.2 SIFT vs LBP vs DAISY vs Normal

La segunda parte de este experimento establece una comparativa de los diferentes detectores de características aplicados a imágenes multiespectrales.

En la siguiente figura 10 se puede ver los diferentes detectores de características y como varían al escoger un entrenamiento con cada imagen completa o con cada pixel o con cada características de cada punto como muestras separadas. Todas las pruebas están hechas con áreas de 40 por 40 alrededor del centro de la imagen en las 15 bandas sin aplicar ninguna transformación a otro espacio de color, y los

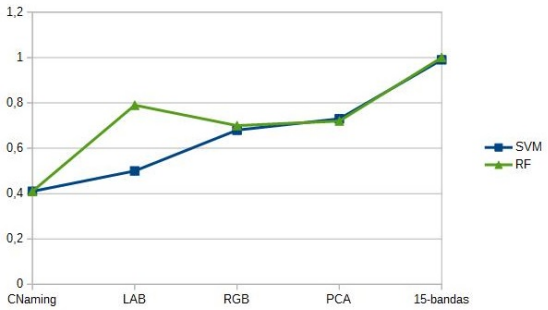

Figure 9: Comparativa de los diferentes espacios de color

descriptores se utilizan con una configuración de un radio de 8 pixeles alrededor de cada punto. En este caso se puede observar que los mejores resultados son los que solo hacen una clasificación por color sin características. Referente al usar pixel a pixel como muestras se puede ver que aúnque hay un cierto empeoramiento este no impide que la clasificación tenga una precisión alta, como es el caso de LBP que con las muestras pixel a pixel pierde usando SVM pero RF es capaz de adaptarse y conseguir un resultado similar al de usar toda la imagen.

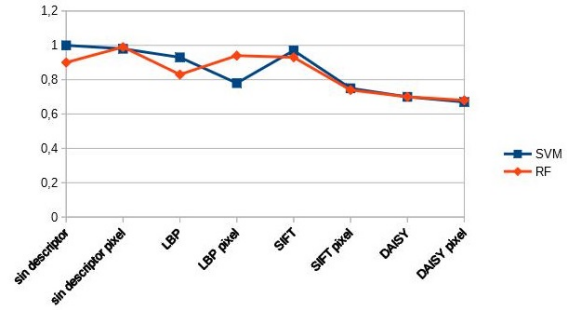

Figure 10: Comparativa de los diferentes descriptores de características

7.3.3 Redes Neuronales

Escogiendo el mismo set de imágenes que en los anteriores apartados, las cuatro panas, se ha entrenado una red neuronal para hacer la clasificación. En este caso, dado el límite de muestras, se han tenido que aplicar diferentes restricciones y técnicas para poder usar este algoritmo de clasificación.

El primer problema que surge de tener un número reducido de muestras es que no se puede entrenar bien una red neuronal desde cero. Por tanto se ha usado una red ya entrenado y se le ha aplicado la técnica de fine-tuning para adaptarla a nuestro problema. La red usada a sido alexnet en el entorno Matconvnet de Matlab. Esta red cuenta con 21 capas de las cuales se ha cambiado la última para poder hacer una clasificación de las 4 tipos de muestras y además se ha aumentado el factor de aprendizaje de las últimas capas convolutivas para poder darle mas libertad para aprender.[14] Por tal de que todos los nodos sean usado y aprendan más o menos el mismo volumen de datos se ha añadido capas de dropout después de las convolutivas para

que el aprendizaje no se centre en unos nodos concretos. [15]

Una de la limitaciones que produce el usar esta red es que su capa de entrada es de 273x273x3 pixeles, o múltiples, esto produce que no se puedan usar imágenes multiespectrales a menos que no se quiera reentrenar de nuevo toda la red, por tanto las comparativas se han reducido al caso del RGB contra la PCA de tres bandas. Aunque no necesita un número de muestras tan elevado como entrenar una red desde cero aún necesita mas imágenes de las que disponemos. Por tal de solucionar esto se le ha aplicado data augmentation al conjunto de muestras, se han usado flips, rotaciones, desenfoque y escalados a cada imagen de cada clase.

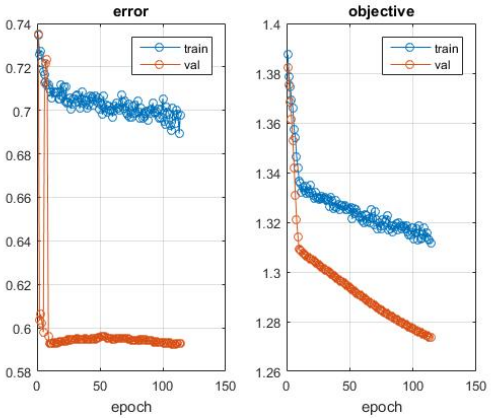

Figure 11: Evolución de la red neuronal en 11 horas

	Black	Blue	Khaki	Wood
Black	5	0	8061	2
Blue	198	0	7694	74
Khaki	0	0	8093	0
Wood	0	0	7962	0

Table 5: RESULTADO DE LA RED NEURONAL

Los primeros resultados de la red neuronal no han sido muy buenos ya que la red extraía precisiones bajas o no clasificaba y catalogaba todas las telas en la misma clase. Como tanto el caso RGB y el caso PCA han extraído resultados idénticos explicaremos solo un ejemplo. En la gráfica 11 se puede observar el error y el objetivo de la red en los 114 epochs que ha realizado en 11 horas. Como se puede ver en pocos epochs consigue un resultado de error que mantendrá a lo largo del resto de iteraciones sin una mejora notable. En la gráfica del objetivo si que se puede observar una mejora durante todas las iteraciones sin aparentemente llegar a un mínimo. Este comportamiento se ha visto en todas las pruebas y en los dos diferentes espacios de colores.

El resultado de esta red con el conjunto de prueba se puede ver en la tabla 5 que representa la matriz de confusión. Como se puede observar ha catalogado todas como Khaki y solo en unos pocos casos a clasificado como Black, pero la mayoría son errónea. Las probabilidades de que una muestra pertenezca a una clase son siempre similares, como ejemplo en el caso de una muestra de la clase Black los resultados son: 0.26 de pertenecer a la clase black, 0.18 de pertenecer a la clase blue, 0.34 de pertenecer a la clase khaki y 0.22 de pertenecer a la clase wood. Como se puede ver la red no acaba de diferenciar de forma significativa las clases.

La explicación de estos resultados puede ser porque todas la telas usadas tienen la misma textura y alexnet es una red que esta entrenada para clasificar imágenes de diferentes objetos sin tener en cuenta la diferencia de color entre las mismas figuras. Como la separación de conjuntos está hecha al azar, por tal de simular un caso real en que no se suele tener el mismo número de muestras de cada clase, puede salir beneficiada alguna clase en la fase de entrenamiento debido a que hay más muestras de una clase que de las otras. Finalmente otro problema es que ha sido entrenado con 11.000 muestras para clase pero estas han sido conseguidas de hacer data augmentation a 30 imágenes de cada clase y como estas no tienen un color uniforme cada pequeñas áreas no tiene porque ser similar a las de su propia imagen creando un cierto grado de confusión.

8 PROBLEMAS ENCONTRADOS

Un primer problema fue que ni Python ni Matlab reconocían la cámara por lo que se tuvo que desarrollar el programa separado de los clasificadores. El lenguaje a usar fue C++ porque la api de la cámara solo acepta C, C++, C# y .NET. Se eligió específicamente C++ debido a que es un lenguaje de alto nivel con orientación a objetos y su facilidad para trabajar en diferentes plataformas adaptando solo algunas partes del código o cambiando algunas librerías.

Después de desarrollar la captura de imágenes hubo problemas en como guardar toda la información en una sola imagen multiespectral. Como no es un campo muy extendido actualmente hay pocos formatos complejos que acepten este tipo de imágenes y algunos suelen tener limitación. Después de hacer un estudio sobre varios tipos de formatos de imágenes multiespectrales tales como GEOTIFF[16], ENVI[17], DICOM[18] o otras opciones de TIFF, se decidió usar TIFF con multisamplingperpixel. Esta opción permite definir varias bandas por pixel siendo 3 bandas el RGB o 1 banda una imagen en escala de grises. La decisión de usar este formato fue porque hay pocas aplicaciones de código libre que guarden en multiespectral o lo reconozcan pero TIFF es un formato que suelen aceptar. En cuanto a la opción se eligió el multisampling porque posteriormente se debía usar un lenguaje de programación que pudiera leer estas imágenes y tanto Python como Matlab no podían leer TIFF multipage, que fue la primera opción que se probo. Al final este problema también produjo que se tuviera que desarrollar los clasificadores con Matlab ya que las librerías de python como OpneCV o Pillow tienen limitaciones a la hora de leer imágenes multiespectrales, en el caso de OpenCV limita a 3 el número de bandas máximas y Pillow lo limita a 11.

Otro problema encontrado es el limitado número de muestras ya que se tarda un tiempo en tomar las muestras y no se puede hacer de forma automática por lo que el número de éstas es muy reducido para aplicar ciertos algoritmos de clasificación, como las redes neuronales,y además no se podía tener confianza en el resultado extraído. Por tal de solventar este problema se optó por usar data augmetation para ampliar el número de muestras y también se usó una

red neuronal preentrenada por tal de reducir el número de muestras y el tiempo requerido para entrenarla. Esto significa que la red neuronal se entrena con muestras de 3 canales por que no se encontró una red multiespectral o suficientemente flexible para poder hacer este tipo de experimentos.

9 Conclusiones

En este proyecto se ha conseguido construir una cámara multiespectral. También se ha evaluado la validez de las fotografías captadas ante diversos clasificadores. La planificación se ha seguido dado que la parte mas crítica está planeada más concretamente, fases mejor definidas y especificadas. En cambio los experimentos con clasificadores ya se previó que habría una dependencia con los resultados de los primeros experimentos por tanto no se concretó fases y se pudo ir cambiando los experimentos según dependiese. Dado que el objetivo del proyecto era el de desarrollar un prototipo y probar su funcionamiento es comprensible tener esta flexibilidad en la etapa de experimentos.

En lo referente a los resultados personales, he podido aprender como usar diversos clasificadores en un entorno poco desarrollado como son las imágenes multiespectrales. También he podido enfrentarme a problemas imprevistos en el desarrollo de un proyecto y el como resolverlo, siempre que se pueda.

En el caso de las imágenes multiespectrales, se ha podido observar que solo aportan más información que las imágenes RGB pero también aumentan el número de características haciendo que sea mas lento el tratamiento de los datos. En el caso de esta cámara se ha encontrado algunos fallos que se puede mejorar, como son la distribución de los puntos de luz y el área que abarcan a iluminar.

Los objetivos principales se han podido cumplir en su mayoría y los problemas se han solucionado en la medida de lo posible. El único objetivo que no se ha podido cumplir, por limitaciones del número de muestras y la complejidad del problema, ha sido el de intentar clasificar piedras de riñón. El objetivo de captar la imagen correctamente se ha podido cumplir pero el resultante no es lo suficientemente bueno debido a que no se puede calibrar correctamente.

10 Líneas futuras

Una propuesta como mejora es la de colocar más LED del mismo tipo en una distribución triangular por tal de eliminar el efecto de sombra de la cámara y poder calibrar la cámara para obtener una imagen más uniforme. También se puede traspasar el control del disparador de la cámara al arduino por tal de ganar velocidad en la captura. Una última mejora que se le puede aplicar a la cámara es substituir la cámara interna de blanco y negro por una de color y poder ampliar el rango de captura de espectros propio de la cámara al poder aprovechar la diferencia entre los LED rojos, verdes i azul de este tipo de cámaras.

Respecto a los experimentos las únicas mejoras que se podrían plantear es aumentar el número de muestras reales sin data augmentation y aumentar la variedad de estas. También se podría entrenar una red neuronal desde cero si se desea usar la cámara para clasificar unicamente por color porque este es un problema muy concreto que depende de las especificaciones de la cámara como es el número de bandas.

Agradecimientos

Este trabajo ha sido sufragado en parte por el proyecto VALTEC 13-1-0148 de la Generalitat de Catalunya.

Me gustaría agradecer a Felipe Lumbreras,como tutor del trabajo, por su apoyo y su soporte en la realización de el Treball de Fi de Grau. Gracias a su guía he podido planificar y comprender problemas que han ido sucediendo y como resolverlos. También agradecer el permitirme experimentar con nuevas tecnologías y ampliar mis conocimientos sobre estas.

También agradezco a mi familia y amigos, por su apoyo y comprensión durante la realización del proyecto. Me gustaría mencionar especialmente a Javier Rodriguez Carmona por sus consejos respecto a las redes neuronales y el como solucionar los problemas que podían ocurrir.

References

[1] M. James. Scrum methodology. [Online]. Available: http://scrummethodology.com/

[2] ——. Basler dart daa2500-14um. [Online]. Available: http://www.baslerweb.com/en/products/cameras/area-scan-cameras/dart/daa2500-14um

[3] S. Leffer. Libtiff-tiff library and utilities. [Online]. Available: http://www.remotesensing.org/libtiff/

[4] Resonon. Hyperespectral imaging applications. [Online]. Available: http://www.resonon.com/applications_main.html

[5] G. Coltof. Hyperspectral techniques explained. [Online]. Available: http://www.bodkindesign.com/wp-content/uploads/2012/09/Hyperspectral-1011.pdf

[6] C. S. Dimitrios Siganos. Neural nteworks. [Online]. Available: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#WhatisaNeuralNetwork

[7] L. Lehe. Principal component analysis. [Online]. Available: http://setosa.io/ev/principal-component-analysis/

[8] OpenCv. Introduction to support vector machines. [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

[9] S. Learn. Decision trees. [Online]. Available: http://scikit-learn.org/stable/modules/tree.html

[10] A. C. Leo Breiman. Random forest. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[11] G. Zhao. Lbp matlab. [Online]. Available: http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

[12] S. Paris. Scenes/objects classification toolbox. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/29800-scenesobjects-classification-toolbox/content/reco_toolbox/html/demo_denseSIFT.html

[13] P. F. Engin Tola, Vincent Lepetit. Daisy: A fast local descriptor for dense matching. [Online]. Available: http://cvlab.epfl.ch/software/daisy

[14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[16] F. Warmerdam. Geotiff. [Online]. Available: http://trac.osgeo.org/geotiff/

[17] H. G. Solutions. Envi header files. [Online]. Available: http://www.harrisgeospatial.com/docs/enviimagefiles.html

[18] NEMA. Dicom. [Online]. Available: http://dicom.nema.org/

APÉNDICE

A.1 RIESGOS

!

Riesgo	Causa	Probabilidad	Impacto	Solución
Mal funcionamiento de la cámara	La cámara no cumple con los requisitos previstos	Baja	Critico	Buscar otra cámara o una mejora del software por tal de poder usarla
No hay distinción en las bandas multiespectral seleccionadas.	Las bandas elegidas no son las correctas.	Media	Medio	Cambiar los LED por otros que sean más adecuados.
No disponer de suficiente material para clasificar	Debido a que todas las imágenes son captadas con una hardware nuevo se puede dar el caso de que no hayan las suficientes para hacer una clasificación.	Alta	Medio	Obtener más imágenes o crear imágenes falsas para poder hacer el experimento
Problemas con la gestión del proyecto, falta de tiempo o mala documentación.	Sucede cuando se dada una mala gestión por la inexperiencia o la mala planificación.	Media	Bajo.	En el momento de detectar este fallo, replantear la planificación y mejorar el control del trabajo realizado.

Table 6: RIESGOS DEL PROYECTO

A.2 Diagrama de Gantt

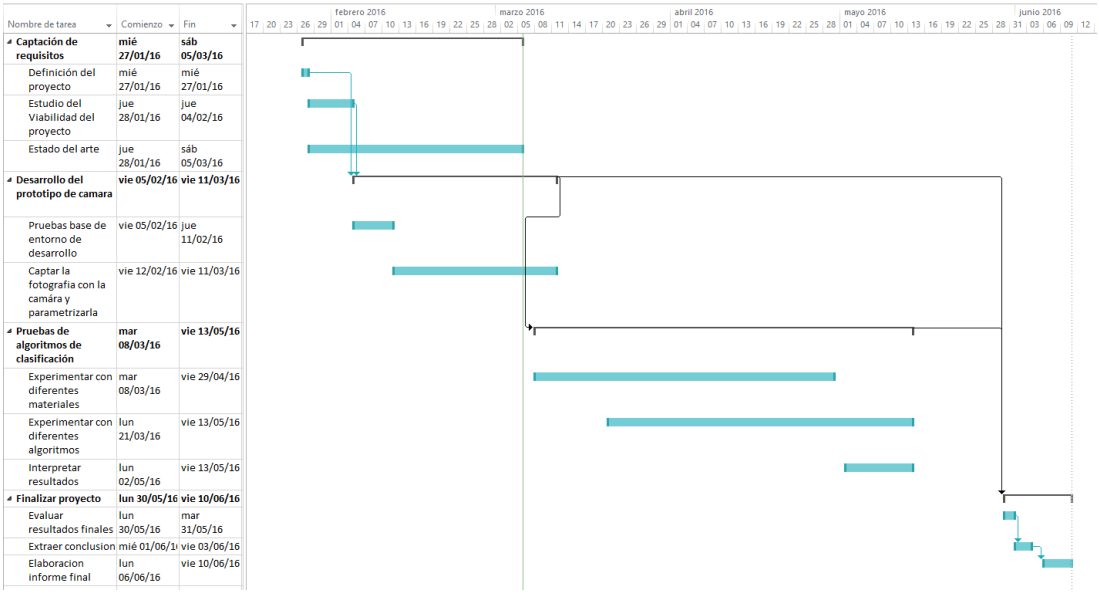

Figure 12: Diagrama de Gantt